UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/665,444	09/18/2003	Marcin Sawicki	60001.0279US01/MS# 304273	7304

7590 02/24/2006

Leonard J. Hope, Esq
Merchant & Gould P.C.
P.O. Box 2903
Minneapolis, MN 55402-0903

EXAMINER	
TRAN, QUOC A	
ART UNIT	PAPER NUMBER
2176	

DATE MAILED: 02/24/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 10/665,444	**Applicant(s)** SAWICKI ET AL.	
	Examiner Quoc A. Tran	**Art Unit** 2176	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *18 September 2003*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-24* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-24* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date *02/02/2004*.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: _____.

DETAILED ACTION

1. This action is responsive to Application filed on 09/18/2003.

2. Claims 1-24 are pending. Claims 1and 10 are independent claims.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> ***(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.***

8. **Claims 1-19 and 23-24** are rejected under 35 U.S.C. 103(a) as being unpatentable over Shaughnessy et al. US 20040205644A1 - filed 12/29/2000 (hereinafter Shaughnessy), in view of Paolo et al. "XLinkProxy: External Linkbases with Xlink" Published by Department of Computer Science University of Bologna, Bologna Italy 11/8-9/2002 (hereinafter Paolo).

In regard to independent claim 1, associating a placeholder with one or more of the markup language tags (Shaughnessy at page 2, paragraph [0037] through page 3 paragraph [0069]), discloses a method and system for creating documents from within a place in collaboration space, upon user selection of the editor selection control, automatically launching a corresponding editor for editing the document; and upon closing the editor, loading the document to place, wherein QuickPlace is representing the collaboration space, which utilizes HTML editor and QP tags, and external files written using Java, and MS office documents from MS office, and further (Shaughnessy at page 30, paragraph [0520] through page 32 paragraph [0544]), discloses the QuickPlace Upload Control Example, The example form of Table 54 creates a page that includes the QuickPlace control to upload files to the QuickPlace. It also includes some fields for the user to fill in to provide information about the attached file. The <body> tag contains the complete form, and the <form> tags are omitted from the HTML page. Adding fields begins right after the <body> tag. Standard HTML fields can be used in the form, and this examples shows use of text fields, a text area and a drop-down field,

Examiner read the above in the broadest reasonable interpretation to the claim limitation; wherein a placeholder with one or more of the markup language tags would have been an obvious variant of QuickPlace, which a place in collaboration space, upon user selection of the editor selection control, automatically launching a corresponding editor for editing the document; and upon closing the editor, loading the document to place, utilizes HTML editor and QP tags, to a person of ordinary skill in the art at the time the invention was made,

displaying the placeholder for each of the one or more markup language tags (Shaughnessy at page 3, paragraph [0073] through page 32 paragraph [0544], also see Fig. 4, 14 and 22-27) discloses a CSOM (Collaboration Space Object Model) preferred to as QuickPlace that includes HTML tags and QPtags, wherein There exists a place 172 that has rooms 174, and there are pages 182 in those rooms. And then there are members 190 of the place. Those four objects 172, 174, 182 and 190 are the primary objects, which are discloses in detail by (Shaughnessy at page 30, paragraph [0520] through page 32 paragraph [0544]), wherein discloses the examples of the use of QP for controlling the form (text fields, text area and a drop-down fields) using QP tags replacing HTML tags,

As illustrates above in Fig. 4 and 14, showing UI (User Interface) for customizing a newly created form, wherein Referring to FIG. 14, the "New Form" scene 430 appears. If

Microsoft Office or Lotus SmartSuite is installed on the user's machine, an additional paragraph 432 is displayed "Based on a Microsoft Office or Lotus SmartSuite Document". This allows the user to create a form based on a Word, Excel, or another Office or SmartSuite document. When pages are created based on this form, the corresponding application will be opened with a particular file. A hidden ActiveX control returns whether any Office applications are installed, or whether any SmartSuite applications are installed. If none are, the upload control 434 and its introduction text 432 are displayed.

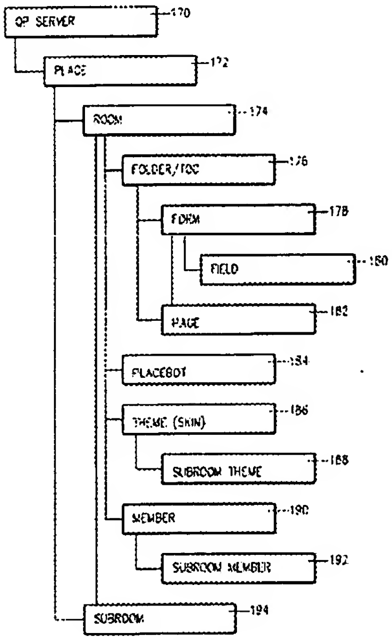
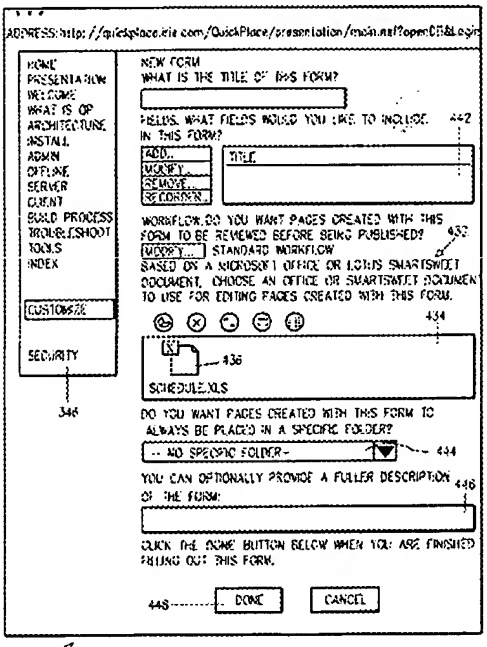


FIG. 4


ADDRESS: http: //quickplace.iris.com/QuickPlace/presentation/main.nsf?openDB&Login

NEW FORM

WHAT IS THE TITLE OF THIS FORM?

FIELDS. WHAT FIELDS WOULD YOU LIKE TO INCLUDE IN THIS FORM?

TITLE

WORKFLOW. DO YOU WANT PAGES CREATED WITH THIS FORM TO BE REVIEWED BEFORE BEING PUBLISHED?

MODIFY... STANDARD WORKFLOW

BASED ON A MICROSOFT OFFICE OR LOTUS SMARTSUITE DOCUMENT. CHOOSE AN OFFICE OR SMARTSUITE DOCUMENT TO USE FOR EDITING PAGES CREATED WITH THIS FORM.

SCHEDULE.XLS

DO YOU WANT PAGES CREATED WITH THIS FORM TO ALWAYS BE PLACED IN A SPECIFIC FOLDER?

-- NO SPECIFIC FOLDER -

YOU CAN OPTIONALLY PROVIDE A FULLER DESCRIPTION OF THE FORM:

CLICK THE DONE BUTTON BELOW WHEN YOU ARE FINISHED FILLING OUT THIS FORM.

FIG. 14

Examiner read the above in the broadest reasonable interpretation to the claim limitation; wherein a placeholder with one or more of the markup language tags that is empty while operating in an editing mode would have been an obvious variant of QuickPlace, which a place in collaboration space, upon user selection of the editor selection control, automatically launching a corresponding editor for editing the document; and upon closing the editor, loading the document to place, utilizes HTML editor and QP tags, and the examples of the use of QP for controlling the form (text fields, text area and a drop-down fields) to a person of ordinary skill in the art at the time the invention was made,

that is empty while operating in an editing mode (Shaughnessy at page 14, paragraph [0215] through page 16 paragraph [0234]) discloses a CSOM (Collaboration Space Object Model) preferred to as QuickPlace that includes HTML tags and QPtags, wherein creating and designing forms is provided, using QuickPlace and interacting with the QuickPlace user interface, create a form as an object of the QuickPlace and select and create fields for that form. For example for empty component using HTML such as,

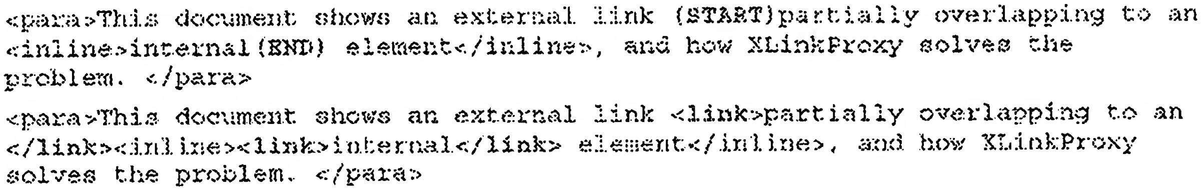
(i) emptyFormat= “”

(ii) prefixHTML= “&lrttr>&lttd>”

(iii) postfixHTML= “</td></tr>”

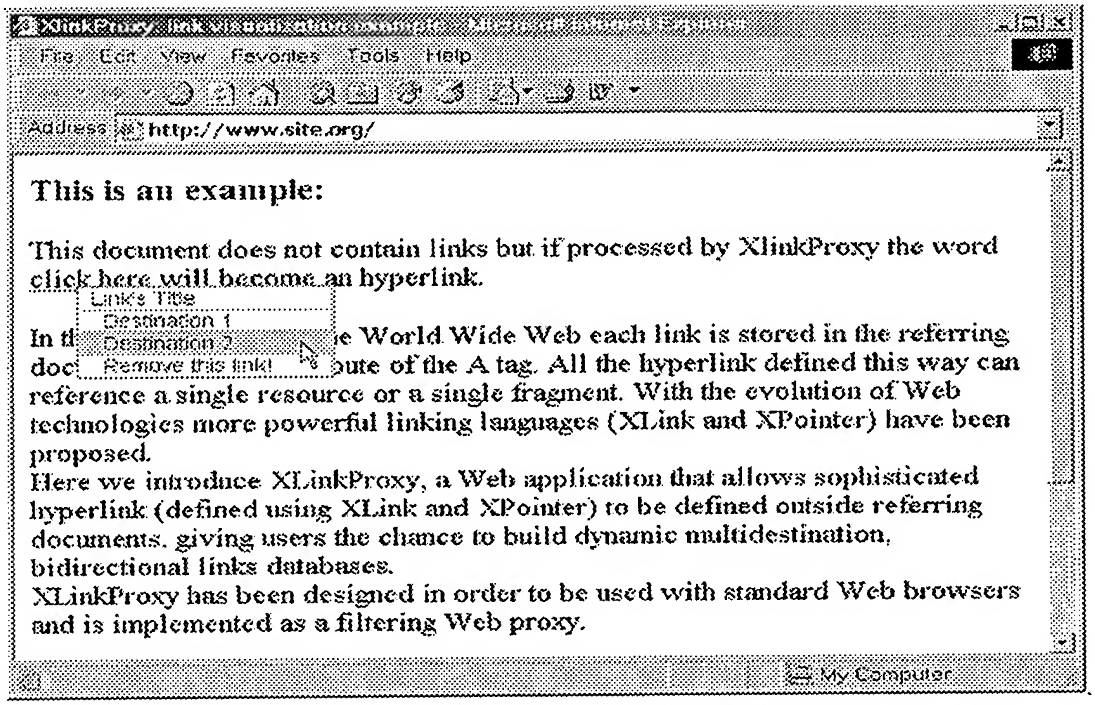
The above sample code would have placed each component in a separate table row, the component's row "collapse" when it is empty, so that it occupies no space. Given that the prefixHTML and postfixHTML parameters are not output when the component is empty, these parameters can be used to provide the following table structure: (see Tables 19-27 for CSS selector and Description and Notes of each selector) particular the Text and fields in page layout in collaborating with (i), (ii) and (iii) for determining the appropriate location for the text and fields and so on…

Shaughnessy does not explicitly teach, **displaying the placeholder for each of the one or more markup language tags that is empty while operating in an editing mode in which the markup language tags are not displayed**, however (Paolo at pages 57-64, also see Fig. 6-8) discloses linking model wherein each link is stored in the referring document within an attribute of the A tag resulting from using XLink and XPointer, The process composes of 8 steps (see page 60-64 and Fig. 4-8) such as the example code shows here:



```
<para>This document shows an external link (START)partially overlapping to an
<inline>internal(END) element</inline>, and how XLinkProxy solves the
problem. </para>
<para>This document shows an external link <link>partially overlapping to an
</link><inline><link>internal</link> element</inline>, and how XLinkProxy
solves the problem. </para>
```

wherein the overlapping anchors often happens that two external links refer to partially overlapping text fragments, or that an external link refers to a text fragment which is partially overlapped by another element in the document, as shown in fig. 4a. In this case it is necessary that the overlap is resolved, possibly by creating multiple fragments for the anchor, before actually inserting the anchor in the document, as shown in fig. 4b. Also illustrating the Nested links, an HTML document with a pop-up for multiple destination links and shown in Fig. 8, an independent frame of the interface, wherein links are added by making selections on the main text, and adding them to either the start points or the end points list. A simple Javascript calculates the correct XPointer. It is then possible to send the link to the XLinkProxy server, provided a linkbase is selected for the link to be stored in. The XPointer uses both the selected string and the position of the string within the document,



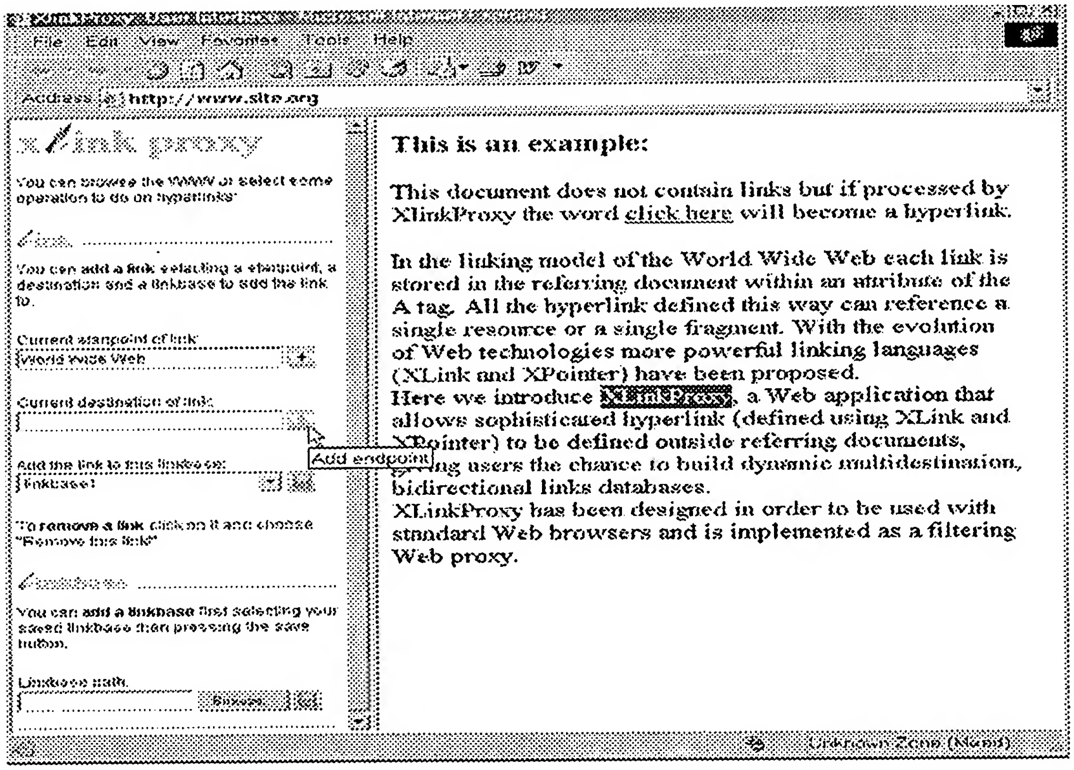
http://www.site.org/

This is an example:

This document does not contain links but if processed by XlinkProxy the word click here will become an hyperlink.

In tl e World Wide Web each link is stored in the referring doc bute of the A tag. All the hyperlink defined this way can reference a single resource or a single fragment. With the evolution of Web technologies more powerful linking languages (XLink and XPointer) have been proposed.
Here we introduce XLinkProxy, a Web application that allows sophisticated hyperlink (defined using XLink and XPointer) to be defined outside referring documents, giving users the chance to build dynamic multidestination, bidirectional links databases.
XLinkProxy has been designed in order to be used with standard Web browsers and is implemented as a filtering Web proxy.

fig. 6: an HTML document with a pop-up menu for a multiple destination link.



http://www.site.org

You can browse the WWW or select some operation to do on hyperlinks

You can add a link selecting a startpoint, a destination and a linkbase to add the link to.

Current startpoint of link:
World Wide Web

Current destination of link:

Add the link to this linkbase:
linkbase1

To remove a link click on it and choose "Remove this link"

You can add a linkbase first selecting your saved linkbase then pressing the save button.

Linkbase path:

This is an example:

This document does not contain links but if processed by XlinkProxy the word click here will become a hyperlink.

In the linking model of the World Wide Web each link is stored in the referring document within an attribute of the A tag. All the hyperlink defined this way can reference a single resource or a single fragment. With the evolution of Web technologies more powerful linking languages (XLink and XPointer) have been proposed.
Here we introduce XLinkProxy, a Web application that allows sophisticated hyperlink (defined using XLink and XPointer) to be defined outside referring documents, giving users the chance to build dynamic multidestination, bidirectional links databases.
XLinkProxy has been designed in order to be used with standard Web browsers and is implemented as a filtering Web proxy.

Fig. 8: the frame used for adding links

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Shaughnessy's teaching, provides associating a placeholder with one or more of the markup language tags and displaying the placeholder for each of the one or more markup language tags that is empty while operating in an editing mode, to include a means of displaying the placeholder for each of the one or more markup language tags that is empty while operating in an editing mode in which the markup language tags are not displayed of Paolo 's teaching. One of ordinary skill in the art would have been motivated to modify this combination to provided a collaboration space application model for creating web applications that are aesthetically pleasing and present the user with a simple interface, that are instantly created, instantly archived, team and project oriented, easy to use, created, accessed and administered via the Web, reusable, and extensible, and enabling creation and editing of documents using legacy editor applications, and for converting the resulting documents to html and uploading them to a place in collaboration space as application enabled files (as taught by Shaughnessy at page 32 paragraphs [0542]-[0544]).

In regard to independent claim 10, incorporates substantially similar subject matter as cited in claim 1 above,. and further in view of the following and therefore is similarly rejected along the same rationale,

Examiner read **second editing mode** in the broadest reasonable interpretation to the claim limitation; wherein **second editing mode** would have been an obvious variant of a place holder discloses in claim 1 which cited above.

.... determined whether any of the tags are empty (Shaughnessy at page 14, paragraph [0215] through page 16 paragraph [0234]) discloses a CSOM (Collaboration Space Object

Model) preferred to as QuickPlace that includes HTML tags and QPtags, wherein creating and designing forms is provided, using QuickPlace and interacting with the QuickPlace user interface, create a form as an object of the QuickPlace and select and create fields for that form. For example for empty component using HTML such as,

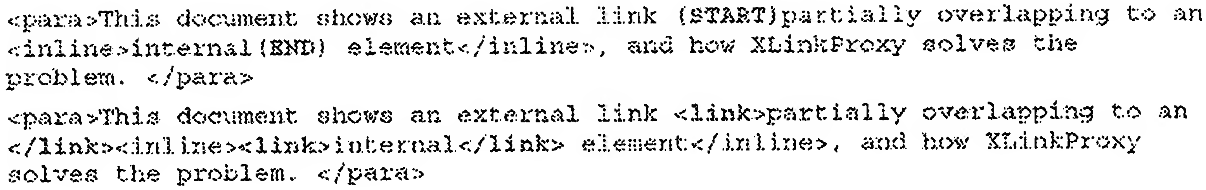
(i) emptyFormat= ""

(ii) prefixHTML= "&lrttr>&lttd>"

(iii) postfixHTML= "</td></tr>"

The above sample code would have placed each component in a separate table row, the component's row "collapse" when it is empty, so that it occupies no space. Given that the prefixHTML and postfixHTML parameters are not output when the component is empty, these parameters can be used to provide the following table structure: (see Tables 19-27 for CSS selector and Description and Notes of each selector) particular the Text and fields in page layout in collaborating with (i), (ii) and (iii) for determining the appropriate location for the text and fields and so on...

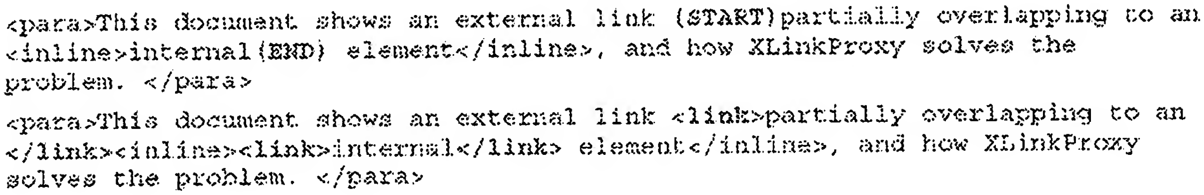
Shaughnessy does not explicitly teach, **providing a first editing mode in which one or more of the markup language tags are displayed,** however (Paolo at pages 57-64, also see Fig. 6-8) discloses linking model wherein each link is stored in the referring document within an attribute of the A tag resulting from using XLink and XPointer, The process composes of 8 steps (see page 60-64 and Fig. 4-8) such as the example code shows here:



```
<para>This document shows an external link (START)partially overlapping to an
<inline>internal(END) element</inline>, and how XLinkProxy solves the
problem. </para>
<para>This document shows an external link <link>partially overlapping to an
</link><inline><link>internal</link> element</inline>, and how XLinkProxy
solves the problem. </para>
```

wherein the overlapping anchors often happens that two external links refer to partially overlapping text fragments, or that an external link refers to a text fragment which is partially overlapped by another element in the document, as shown in fig. 4a. In this case it is necessary that the overlap is resolved, possibly by creating multiple fragments for the anchor, before actually inserting the anchor in the document, as shown in fig. 4b,

Examiner read the above in the broadest reasonable interpretation to the claim limitation; wherein the first editing mode would have been an obvious variant of



```
<para>This document shows an external link (START)partially overlapping to an
<inline>internal(END) element</inline>, and how XLinkProxy solves the
problem. </para>
<para>This document shows an external link <link>partially overlapping to an
</link><inline><link>internal</link> element</inline>, and how XLinkProxy
solves the problem. </para>
```

to a person of ordinary skill in the art at the time the invention was made,

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Shaughnessy's teaching, provides associating a placeholder with one or more of the markup language tags and displaying the placeholder for each of the one or more markup language tags that is empty while operating in an editing mode, to include a means of displaying the placeholder for each of the one or more markup language tags that is empty while operating in an editing mode in which the markup language tags are not displayed and providing a first editing mode in which one or more of the markup language tags are displayed, of Paolo 's teaching. One of ordinary skill in the art would have been motivated to modify this combination to provided a collaboration space application model for creating web applications that are aesthetically pleasing and present the user with a simple interface, that are instantly created, instantly archived, team and project oriented, easy to use, created, accessed and administered via the Web, reusable, and extensible, and enabling creation and editing of documents using legacy editor applications, and for converting the resulting documents to html and uploading them to a place in collaboration space as application enabled files (as taught by Shaughnessy at page 32 paragraphs [0542]-[0544]).

In regard to dependent claim 2, wherein the placeholder for each tag comprises the name of the tag (Shaughnessy at page 14, paragraph [0215] through page 16 paragraph [0234]).

In regard to dependent claim 3, wherein the placeholder for each tag comprises a user-defined text string (Shaughnessy at page 14, paragraph [0215] through page 16 paragraph [0234]).

In regard to dependent claim 4, wherein the placeholder for each tag comprises a user-defined text string or the name of the tag if a user-defined text string has not been defined for the tag (Shaughnessy at page 14, paragraph [0215] through page 16 paragraph [0234]).

In regard to dependent claim 5, incorporates substantially similar subject matter as cited in claims 1 and 10, and is similarly rejected along the same rationale.

In regard to dependent claim 6, incorporates substantially similar subject matter as cited in claims 1, 6 and 10, and further view of the following, and are similarly rejected along the same rationale,

in response to the typed data, removing the place holder for the tag and inserting the typed data within the tag, (Shaughnessy at page 30, paragraph [0516] through page 32 paragraph [544]) discloses toolbar in Word offers eleven standard fields to use in a document: Checkbox control, Option or Radio button control, Drop Down box control, Listbox control, Textbox control, Text Area control, Submit control, Submit with Image control, Reset control, Hidden control, and Password control, By selecting control and clicking on Properties, an HTML name and value can be assigned to the field, such the sample code as followings:

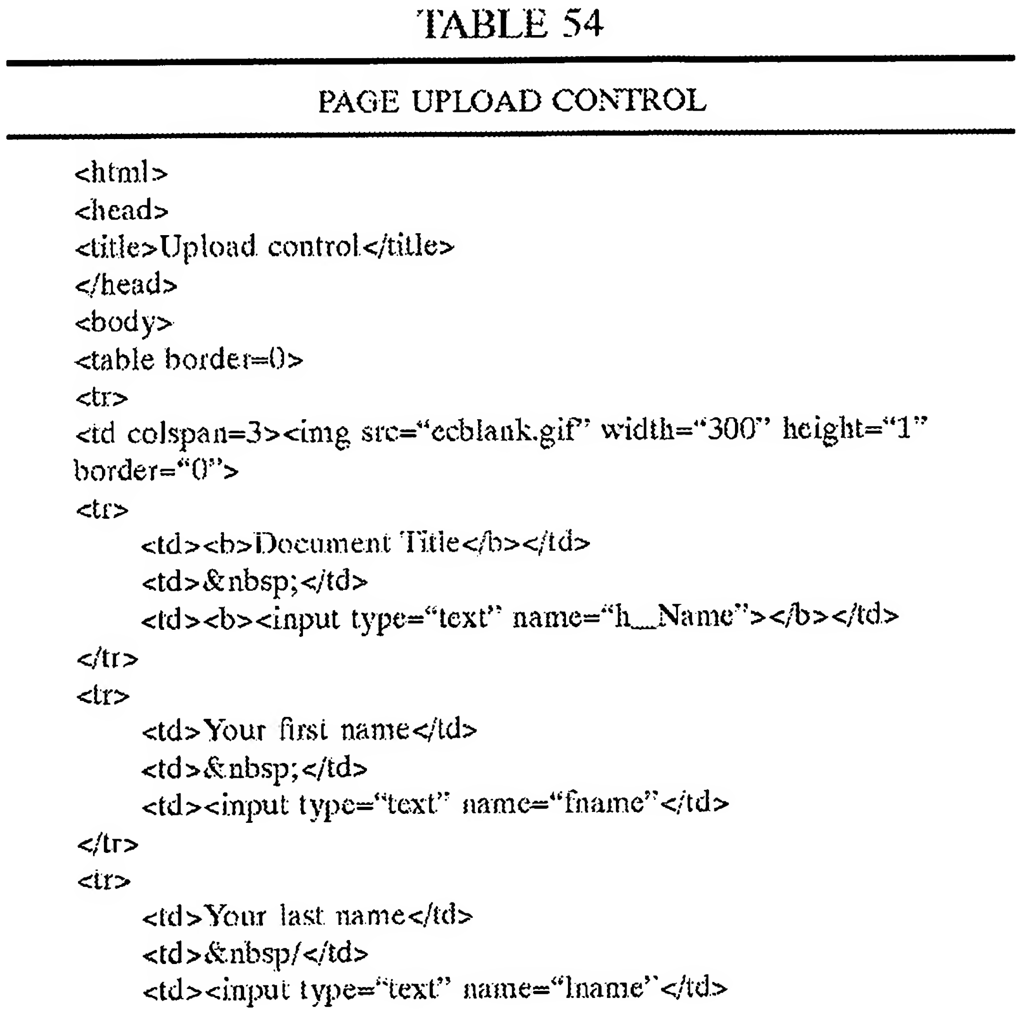


TABLE 54

PAGE UPLOAD CONTROL

```
<html>
<head>
<title>Upload control</title>
</head>
<body>
<table border=0>
<tr>
<td colspan=3><img src="ecblank.gif' width="300" height="1"
border="0">
<tr>
     <td><b>Document Title</b></td>
     <td>&nbsp;</td>
     <td><b><input type="text" name="h_Name"></b></td>
</tr>
<tr>
     <td>Your first name</td>
     <td>&nbsp;</td>
     <td><input type="text" name="fname"</td>
</tr>
<tr>
     <td>Your last name</td>
     <td>&nbsp/</td>
     <td><input type="text" name="lname"</td>
```

Also the QuickPlace component is included for the Rich text control and specific JavaScript (e.g. specific JavaScript functions to use with forms exist in QuickPlace. They are event handlers that can be called when the form is loaded from or a page created by the form is submitted to the QuickPlace), Examiner read the above in the broadest reasonable interpretation to the claim limitation; wherein the typed data and removing the place holder for the tag and inserting the typed data within the tag would have been an obvious variant of Table 54 and the QP component is included for the Rich text control and specific JavaScript (e.g. specific JavaScript functions to use with forms exist in QuickPlace. They are event handlers that can be called when the form is loaded from or a page created by the form is submitted to the QuickPlace), to a person of ordinary skill in the art at the time the invention was made, since the event handlers that can be called when the form is loaded from or a page created by the form is submitted to the QuickPlace and replace by the result from table 54.

In regard to dependent claim 7, incorporates substantially similar subject matter as cited in claims 1, 6 and 10, and is similarly rejected along the same rationale.

In regard to dependent claim 8, is directed toward a computer readable medium having computer-executable instruction for executing the method of claim 1, and is similarly rejected along the same rationale.

In regard to dependent claim 9, is directed toward a computer-control apparatus for performance the method of claim 1, and is similarly rejected along the same rationale.

In regard to dependent claim 11, incorporates substantially similar subject matter as cited in claims 1, 2 and 10, and is similarly rejected along the same rationale.

In regard to dependent claim 12, incorporates substantially similar subject matter as cited in claim 10, and is similarly rejected along the same rationale.

In regard to dependent claim 13, incorporates substantially similar subject matter as cited in claims 1and 6, and is similarly rejected along the same rationale.

In regard to dependent claim 14, incorporates substantially similar subject matter as cited in claims 1, 6 and 10, and is similarly rejected along the same rationale.

In regard to dependent claim 15, incorporates substantially similar subject matter as cited in claims 1 and 10, and is similarly rejected along the same rationale.

In regard to dependent claim 16, incorporates substantially similar subject matter as cited in claims 1, 10 and 15, and is similarly rejected along the same rationale.

In regard to dependent claim 17, wherein the request to remove the tag comprises a drag and drop operation, (Shaughnessy at page 23, paragraph [0352] through page 24 paragraph [0382]), discloses the drag and drop operation.

In regard to dependent claims 18-19, incorporates substantially similar subject matter as cited in claims 1, 8 and 10, and are similarly rejected along the same rationale.

In regard to dependent claim 23, is directed toward a computer readable medium having computer-executable instruction for executing the method of claim 10, and is similarly rejected along the same rationale.

In regard to dependent claim 24, is directed toward a computer-control apparatus for performance the method of claim 10, and is similarly rejected along the same rationale.

9. **Claims 20-22** are rejected under 35 U.S.C. 103(a) as being unpatentable over Shaughnessy et al. US 20040205644A1 - filed 12/29/2000 (hereinafter Shaughnessy), in view of Paolo et al. "XLinkProxy: External Linkbases with Xlink" Published by Department of Computer Science University of Bologna, Bologna Italy 11/8-9/2002 (hereinafter Paolo), further in view of AbiWord Schema at www.abisource.com/awml.xsd published location at

www.w3.org/2OOO/10/XMLSchema and the XSD for AbiWord, (hereinafter AbiWord Schema), further in view of Larry Ayers "AbiWord's Potential" Copyright © 1999, Published in Issue 43 of Linux Gazette, July 1999 pages 1-4 (hereinafter Ayers).

In regard to dependent claims 20-22, Shaughnessy and Paolo do not explicitly teach, **wherein the text is defined in XSD schema file, wherein the text is defined in a schema library wherein the text is defined in a configuration**, however (AbiWord Schema, pages 1-3), discloses the utilizing an XSD, or XML Schema Definition, which represents a word processor's rich formatting, and which is published and available to other applications, for interpreting and validating the word-processor document. AbiWord Schema discloses an XSD for validating an AbiWord document. AbiWord Schema also discloses a definition of AbiWord's rich formatting, including styles, lists, sections and data types (See AbiWord Schema, Page 1, lines 16-19). AbiWord Schema also discloses the location of the published XML Schema at www.w3.org/2OOO/10/XMLSchema and the XSD for AbiWord, published at www.abisource.com/awml.xsd (See AbiWord Schema, Page 1, line 3 and trailer line, bottom of page).

In Addition, further in view of AbiWord' Potential of (Ayers, Pages 1-3), discloses the word-processor document, in the form of creating and examining an AbiWord file (*.abw) that recreates the word processor's set of features. An *.abw file is written in XML and thus is also in ASCII format; the files can be read by any text editor. This is quite a break with word processor tradition and ensures that when you write a document with AbiWord you don't run the risk of being strictly tied to one particular word processor, which may not even run on machines five years from now. AbiWord can also save in the HTML and RTF formats, both of which are accessible with word processors such as MS-Word and WordPerfect. Due to limitations of HTML and RTF some formatting information is lost (such as the specific fonts used), but attributes such as bold and italic font styles and tab-settings are retained. If XML really does become a widely-used and open data-format (as its proponents predict) AbiSource might be in a good position to gain users and clients are available from the AbiSource web-page, http://www.abisource.com/,

Ayers and AbiWord Schema are analogous art, because they are from the same field of endeavor of creating and manipulating AbiWord document. At the time of the invention it would have been obvious to a person of the ordinary skill in the art to include the AbiWord XSD of AbiWord Schema with the AbiWord document of Ayers. The motivation of doing so would have been to validating the AbiWord document (see AbiWord Schema, page 1, lines 8-9). Therefore, it would have been obvious to combine AbiWord Schema with Ayers for benefit of validating an AbiWord document to obtain the document wherein the text is defined in XSD schema file, wherein the text is defined in a schema library wherein the text is defined in a configuration of claims 20-22.

Also It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Shaughnessy teaching, provides associating a placeholder with one or more of the markup language tags and displaying the placeholder for each of the one or more markup language tags that is empty while operating in an editing mode, to include a means of displaying the placeholder for each of the one or more markup language tags that is empty while operating in an editing mode in which the markup language tags are not displayed and providing a first editing mode in which one or more of the markup language tags are displayed, of Paolo 's teaching, further to include the text wherein the text is defined in XSD schema file, and wherein the text is defined in a schema library and wherein the text is defined in a configuration of Ayers and AbiWord Schema teaching. One of ordinary skill in the art would have been motivated to modify this combination to provided a collaboration space application model for creating web applications that are aesthetically pleasing and present the user with a simple interface, that are instantly created, instantly archived, team and project oriented, easy to use, created, accessed and administered via the Web, reusable, and extensible, and enabling creation and editing of documents using legacy editor applications, and for converting the resulting documents to html and uploading them to a place in collaboration space as application enabled files (as taught by Shaughnessy at page 32 paragraphs [0542]-[0544]).

Conclusion

10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Quoc A. Tran whose telephone number is (571) 272-4103. The examiner can normally be reached on Monday through Friday from 9 AM to 5 PM EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Herndon R. Heather can be reached on (571) -272-4136. The fax phone number for the organization where this application or proceeding is assigned is (571)-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Quoc A, Tran
Patent Examiner
Technology Center 2176
February 17, 2006

William L. Bashore
WILLIAM BASHORE
PRIMARY EXAMINER
2/17/2006